Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 7, Number 1 (2011), pp. 101–111
© Research India Publications
http://www.ripublication.com/gjpam.htm

An Alternative Approach to Solve Quadratic Programming Problem with Homogeneous Constraints

Archana Khurana[*,1], Sanjeet Singh[2] and S.R.Arora[3]

[1] *University School of Basic and Applied Sciences, Guru Gobind Singh Indraprastha University, Dwarka Sector-16C, Delhi-110075, India.*
[*] *Corresponding Author E-mail: archana2106@gmail.com*
[2] *Operations Management Group,*
Indian Institute of Management Calcutta, Kolkata-700104, India.
E-mail: sanjeet@iimcal.ac.in
[3] *Department of Mathematics, HansRaj College University of Delhi,*
Delhi-110007, India. E-mail: srarora@yahoo.com

Abstract

In this paper, we have developed an alternative approach to solve quadratic programming problem with homogenous constraints. Our approach is an alternative to the existing variable elimination method to remove homogeneous constraints. Using the homogenous constraint a transformation matrix T is constructed which helps in reducing the given quadratic programming problem into another quadratic programming problem having fewer constraints. A relationship between the original problem and the transformed problem is also established which ensures that the solution of the original problem can be obtained from the transformed problem. Theoretical results are illustrated with the help of a numerical example.

Mathematics subject classification 1991
Primary: 90C20, 90C32**; Secondary:** 90C26

Key Words: Quadratic programming, Homogeneous constraints, Transformation matrix.

Introduction

Quadratic Programming Problem is a special type of mathematical optimization problem, which involves maximization/minimization of a sum of a quadratic and a linear function subject to linear constraints. Quadratic programming problem with homogenous constraints refers to minimizing a sum of quadratic and linear function subject to linear constraints where some of the constraints are homogeneous. The general quadratic programming problem with homogeneous constraints can be written as

$$\text{(P1)} \qquad Maximize\ f(x) = cx + x^tQx$$
$$\text{subject to } Ax = b$$
$$x \geq 0$$

with its i^th constraint as $a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = 0$. In Problem (P1) **c** is an n-dimensional row vector describing the coefficients of the linear terms in the objective function, **Q** is an $(n \times n)$ symmetric matrix describing the coefficients of the quadratic term, $\boldsymbol{x}$ is the n-dimensional column vector of the decision variables, constraints coefficients are defined by an $(m \times n)$ matrix $\boldsymbol{A}$ and $\boldsymbol{b}$ is an m-dimensional column vector of the right-hand-side values. We assume that a feasible solution exists and that the feasible region is bounded. When the objective function $f(x)$ is strictly concave for all feasible points, the problem (P1) has a unique local maximum which is also the global maximum. A sufficient condition to guarantee strictly concavity of f(x) is Q should be negative definite.

Research literature is full of variety of applications of quadratic programming such as portfolio optimization [10], structural analysis [3], optimal control [4], classification [7] and quadratic transportation problems [1, 2, 9]. Also quadratic programming problem with homogeneous constraints occurs in many real life situations, for example, in portfolio optimization when investment in securities in one sector is dependent on the investment in securities in another sector. Various methods for solving a quadratic programming problem have already been developed. Wolf [11] in 1959 gave the Simplex method for solving quadratic programming problem. In 1964 Boot [5] also solved the quadratic program. In 1996 Horst [8] et. al. gave a new algorithm for solving the general quadratic programming problem. Chadda [6] in 1999 developed an algorithm to solve a linear fractional program with homogeneous constraints. In this paper, we have developed an alternative approach to solve quadratic programming problem with homogenous constraints. Our approach is an alternative to the existing variable elimination method to remove homogeneous constraints. Using the homogenous constraint a transformation matrix T is constructed which helps in reducing the given quadratic programming problem into another quadratic programming problem having fewer constraints. A relationship between the original problem and the transformed problem is also established which ensures that the solution of the original problem can be obtained from the transformed problem.

Development of transformation matrix *T*

Let $S_1 = \{x: Ax = b, x \geq 0\}$ denotes the feasible region for (P1). Also let $x = (x_1, x_2, \cdots, x_n)$ be a solution of the problem (P1). It is obvious that if x_k and $a_{ik} > 0$ then there exists at least one x_l with $a_{il} < 0$. Now, corresponding to first homogeneous constraint, let R be a $1 \times n$ row vector and $R = (R_1, R_2, \cdots, R_n)$ where $R_1, R_2, \cdots, R_n$ are the columns of R. So we partition $R = (R^0, R^+, R^-)$ where R^0 is the set of all columns of R for which $a_{ij} = 0, \forall\, i$ let the number of such columns be r; $R^+$is the set of all columns of R for which $a_{ij} > 0, \forall\, i$ let the number of such columns be p; R^- is the set of all columns of R for which $a_{ij} < 0, \forall\, i$ let the number of such columns be q.

Thus $p + q + r = n$. Now we define a transformation matrix T of order $n \times (pq + r)$ such that the i^{th} equation of ATw = b will be identically zero where w is a column vector with $(pq + r)$ components. Partition T as $T = (T_1, T_2)$ where T_1consists of unit column vectors e_j corresponding to $a_{ij} = 0$ and T_2 consists of column vectors corresponding to w_{kl}, where w_{kl} is defined as $(k, l)^{th}$element of w such that $R^k \in R^+$ $k = 1,2, \ldots, p$ and $R^l \in R^-$, $l = 1,2, \ldots, q$. The matrix T possesses n rows and $r + pq$ columns and is represented as $T = (T_1, T_2) = \{(e_j)$ such that $a_{ij} = 0\ \forall$ i ; $(t_{kl})\ \forall\ k \in R^+$ and $l \in R^-\}$ i.e., e_j is the j^{th} column vector of the identity matrix I_n and

$$t_{kl} = -a_{il}e_k + a_{ik}e_l \tag{1}$$

Remark 1. We have constructed the transformation matrix T in such a way that the i^{th} equation of $ATw = b$ is identically zero. Therefore it is sufficient to show that the i^{th} row of AT will have all its elements as zeros. Let A^i be the ith row of A, then for any $j \in A^0$, it is clear that $A^i e_j = 0$which implies that $A^i T_1 = 0$.

$$\begin{aligned} \text{i.e., } A^i t_{kl} &= A^i(-a_{il}e_k + a_{ik}e_l) \text{ for any } (k, l),\ k \in A^+ \text{and } l \in A^- \\ &= -a_{il}a_{ik} + a_{ik}a_{il} = 0\ \forall\ k \in A^+ \text{and } l \in A^- \end{aligned}$$

Hence the elements of the i^{th} row of ATw will have all the zeros. Also for the i^{th} equation $b_i = 0$. Thus the i^{th} equation of $ATw = b$ will be identically zero.

Equivalence relationship

Applying the transformation $x = Tw$ in the original problem (P1), we have the following transformed problem (P2):

$$\text{(P2)} \qquad \text{Maximize } g(w) = \bar{c}w + w^t\overline{Q}w$$
$$\text{subject to } \overline{A}w = b$$
$$w \geq 0$$

where $\overline{A} = AT$, $\overline{c} = cT$ and $\overline{Q} = T^tQT$

Clearly, $\overline{Q}$ will also be a negative definite symmetric matrix.

As the i^{th} equation of $ATw = b$ will be identically zero, it can be removed while solving the problem (P2). Let $S_2 = \{w: \bar{A}w = b, w \geq 0\}$.

Lemma: If $\sum_{i=1}^{p} \alpha_i = \sum_{j=1}^{q} \beta_j = V$, $\alpha_i \geq 0$, $\beta_j \geq 0$, then there exists a matrix y = (y_{ij}) (where $y_{ij} \geq 0 \ \forall \ (i,j)$) such that $\sum_{j=1}^{q} y_{ij} = \alpha_i$ and $\sum_{i=1}^{p} y_{ij} = \beta_j$

Proof: Define $y_{ij} = \dfrac{\alpha_i \times \beta_j}{V}$

Now, $$\sum_{i=1}^{p} y_{ij} = \sum_{i=1}^{p} \frac{\alpha_i \times \beta_j}{V} = \frac{\beta_j \sum_{i=1}^{p} \alpha_i}{V} = \frac{\beta_j \times V}{V} = \beta_j$$

and $$\sum_{j=1}^{q} y_{ij} = \sum_{j=1}^{q} \frac{\alpha_i \times \beta_j}{V} = \frac{\alpha_i \sum_{j=1}^{q} \beta_j}{V} = \frac{\alpha_i \times V}{V} = \alpha_i$$

Also, $y_{ij} \geq 0$ ($\because \alpha_i \geq 0, \beta_j \geq 0$)

Thus the existence of the matrix y= (y_{ij}) (where $y_{ij} \geq 0 \ \forall \ (i,j)$) is ensured.

Theorem 1: If x solves $Ax = b$ then there exists a w, such that $x = Tw$, which solves $\overline{A}$ w = b.

Proof: As x is a solution of $Ax = b$, then its i^{th} constraint equation will be $\sum_{k \in A^+} a_{ik} x_k + \sum_{l \in A^-} a_{il} x_l = 0$

Putting $a_{ik} x_k = \alpha_k \ and \ -a_{il} x_l = \beta_l$ (2)

We have $\sum_{i=1}^{p} \alpha_i = \sum_{j=1}^{q} \beta_j$

Thus by lemma, $\exists$ a matrix y (= $y_{ij} \geq 0$) of order $p \times q$ such that

$$\sum_{l=1}^{p} y_{kl} = \alpha_k \ and \ \sum_{k=1}^{p} y_{kl} = \beta_l \qquad (3)$$

Now we define a vector $w = \begin{bmatrix} w^1 \\ w^2 \end{bmatrix}$ where w^1 is a column vector with r components and w^2 is a column vector with pq components.

Also, $w_j^1 = x_j \quad \forall\, j \in A^0$

and $w_{kl}^2 = \dfrac{-y_{kl}}{a_{ik}a_{il}} \;\forall\, k \in A^+$ and $l \in A^-$ (4)

Clearly, $w \geq 0$ ($\because$ $a_{ik} > 0$, $a_{il} < 0$ and $y_{kl} \geq 0$)

Next we wish to prove that $ATw = b$ which is equivalent to proving that $Tw = x$
Now, $Tw = T^1w^1 + T^2w^2$

$$= \sum_j e_j w_j^1 + \sum_k \sum_l t_{kl} w_{kl}^2$$

$$= \sum_j e_j x_j + \sum_k \sum_l \left(-a_{il}e_k + a_{ik}e_l\right)\left(\frac{-y_{kl}}{a_{ik} \times a_{il}}\right) \qquad \text{(using (1) and (4))}$$

$$= \sum_j e_j x_j + \sum_k \sum_l \frac{e_k a_{il} y_{kl}}{a_{ik}a_{il}} - \sum_k \sum_l \frac{a_{ik} e_l y_{kl}}{a_{ik}a_{il}}$$

$$= \sum_j e_j x_j + \sum_k \frac{e_k}{a_{ik}} \alpha_k - \sum_l \frac{e_l}{a_{il}} \beta_l \qquad \text{(using (3))}$$

$$= \sum_j e_j x_j + \sum_k \frac{e_k}{a_{ik}} a_{ik} x_k - \sum_l \frac{e_l}{a_{il}} (-a_{il} x_l) \qquad \text{(using (2))}$$

$$= \sum_{j \in A^0} e_j x_j + \sum_{k \in A^+} e_k x_k + \sum_{l \in A^-} e_l x_l$$

$= x$
$\Rightarrow Tw = x$
Since $Ax = b$ therefore $ATw = b$
$\Rightarrow \bar{A}w = b \Rightarrow$ w solves $\bar{A}w = b$.

Theorem 2: If x^* solves the problem (P1) then w^* $(x^* = Tw^*)$ solves the problem (P2).

Proof: Since the existence of a feasible solution w^* is guaranteed by theorem 1.
Therefore $Aw^* = b$, $w^* \geq 0$, Now as x^* solves the program (P1).
Thus $cx^* + x^{*t}Qx^* \geq cx + x^tQx.$
Since $x^* = Tw^*$ and $x \quad = Tw$
$\therefore cTw^* + \left(Tw^*\right)^t Q\left(Tw^*\right) \geq cTw + w^tT^tQTw$
$\Rightarrow \bar{c}w^* + w^{*t}\bar{Q}w^* \geq \bar{c}w + w^t\bar{Q}w \;\forall\, w \in S_2$
$\Rightarrow w^*$ solves the problem (P2).

Theorem 3: If w^* solves the problem (P2) then there exists $x^* = Tw^*$ which solves the problem (P1) and the extreme values of the two objective functions are equal.

Proof: Since w^* solves the problem (P2) . Therefore, $\overline{A}w^* = b$, $w^* \geq 0 \Rightarrow ATw^* = b$ i.e., $Ax^* = b$.

Further $T \geq 0$, $w^* \geq 0$ and as $Tw^* = x^*$, we get $x^* \geq 0$

Also we have, $\bar{c}w^* + w^{*t}\,\overline{Q}w^* \geq \bar{c}w + w^t\,\overline{Q}w \;\; \forall\, w \in S_2$ (5)

Let $\bar{x}$ not x^* solves the problem (P2),

then $c\bar{x} + \bar{x}^t Q\bar{x} \geq cx^* + x^{*t} Qx^*$

From theorem 1, it follows that $\overline{w}$ solves the problem (P2).

Hence, $cT\overline{w} + (T\overline{w})^t Q(T\overline{w}) \geq cTw^* + (Tw^*)^t Q(Tw^*)$

$\Rightarrow cT\overline{w} + \overline{w}^t T^t QT\overline{w} \geq cTw^* + w^{*t} T^t QTw^*$

i.e., $\bar{c}\,\overline{w} + \overline{w}^t\,\overline{Q}\,\overline{w} \geq \bar{c}w^* + w^{*t}\,\overline{Q}w^*$, which contradicts (5).

Hence x^* solves the program (P1).

Next let $f(x^*)$ and $g(w^*)$ be the optimal values of the programs (P1) and (P2) at x^* and w^* respectively.

$$\begin{aligned}
\text{Then } f(x^*) &= cx^* + x^{*t} Qx^* \\
&= cTw^* + (Tw^*)^t Q(Tw^*) \\
&= cTw^* + w^{*t} T^t QTw^* \\
&= \bar{c}w^* + w^{*t}\,\overline{Q}w^* \\
&= g(w^*)
\end{aligned}$$

Hence the optimal values of the two objective functions are equal.

Remark 2: Our method may be useful to a large class of programming model containing a huge number of homogeneous constraints. If we have a large number of homogeneous constraints then the variable elimination method may be quite cumbersome as each time we will have to substitute the value of a dependent variable in terms of independent variables in all the equations to eliminate one constraint. The method presented in this paper is an alternative to variable elimination method and removes one constraint at each step by defining a transformation $x = Tw$. However, the method presented in the paper doesn't necessarily decrease the number of variables.

Remark 3: The MATLAB code is also developed for reducing the original problem (P1) to transformed problem (P2) without homogeneous constraint. We find out the transformation matrices through Matlab. Its code is given in appendix A.

Numerical example

Consider Maximize $f(x) = 2x_1 + 3x_2 - x_1^2$

subject to $-x_1 + 2x_2 + x_3 = 2,$

$-x_1 + x_2 + x_4 = 0,$

$-5x_1 + 3x_2 + x_5 = 0,$

$x_j \geq 0$, j = 1,2,…..,5.

The optimal solution of this problem is $x_1=2$, $x_2=2$, $x_3=0$, $x_4=0$, $x_5=4$ with maximum value of the objective function = 10.

The above problem can equivalently be written in the form

Maximize $g(w) = cTw + w^tT^tQTw$

subject to $ATw = b,$

$w \geq 0.$

i.e., Maximize $g(w) = \bar{c}w + w^t\bar{Q}w$

subject to $\bar{A}w = b$,

$w \geq 0.$

Since there are two homogeneous constraints, so there will be two transformation matrices.

Firstly we find out the first transformation matrix T(1) given by

$$T(1) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$ corresponding to the first homogeneous constraint.

This reduces our problem to

Maximize $g(w) = 5w_3 + 2w_4 - w_3^2 - 2w_3w_4 - w_4^2$

subject to $w_1 + w_3 - w_5 = 2,$

$w_2 - 2w_3 - 5w_4 = 0,$

$w_i \geq 0$, i = 1,2,3,4

where w is given by $x = T(1)w$

Since in the transformed problem there is another homogeneous constraint present, so we find the second transformation matrix $T(2)$ given by

$$T(2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which reduces our problem to

Maximize $h(y) = 5y_2 + 2y_3 - y_2^2 - 2y_2y_3 - y_3^2$

subject to $y_1+y_2-y_3 = 2$

$y_2, y_3 \geq 0$

where y is given by $w = T(2)y$

Its optimal solution is $y_1 = 0, y_2 = 2, y_3 = 0$

Hence the solution of the original problem is given by $x = T(1)T(2)y$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \\ 4 \end{bmatrix}$$

Hence the optimal solution of the given problem is $x_1 = 2, x_2 = 2, x_3 = 0, x_4 = 0, x_5 = 4$ with maximum value of $f(x)$ =10.

Conclusion

The process described in section 2 can be extended to define T if $Ax = b$ has more than two homogenous constraints. In case there are s homogenous constraints, we define s transformation matrices $T(1), T(2), \dots\dots, T(s)$. Note that T(2)is determined once $AT(1)$ has been computed. In general, $T(s)$ is determined only when $AT(1)T(2)\dots\dots..T(s-1)$ has been computed. The alternative method developed in this paper transforms the original problem to a new formulation which has lesser number of constraints but more number of variables. Here, it may be noted that as complexity of simplex type algorithms depend more on number of constraint than number of variables. Therefore, alternative method developed in this paper is useful when there are huge number of homogeneous constraints in a quadratic programming problem.

Acknowledgement

The authors are thankful to Mr. Anish Goyal, B.Tech (IT) 3rd year student of Guru Gobind Singh Indraprastha University for helping us in developing the MATLAB code for our problem.

References

[1] Arora S.R. and Khurana A., Three dimensional fixed charge bi-criterion indefinite quadratic transportation problem, Yugoslavia Journal of Operations Research, 2004, 14(1), 83-97.

[2] Arora, S. R. and Khurana, A., A paradox in an indefinite quadratic transportation problem, International Journal of Management Science, 2001, 7(2), 13-30.

[3] Atkociunas J., Quadratic programming for degenerate shakedown problems of bar structures, Mechanics Research Communications, 1996, 23(2), 195-203.
[4] Bashein G. and Enns. M., Computation of optimal controls by a method combining quasi-linearization and quadratic programming, International Journal of Control, 1972, 16(1), 177-187.
[5] Boot J., Quadratic programming, Rand McNally, Chicago, 1964.
[6] Chadha S.S., A linear fractional program with homogeneous constraints, Opsearch, 1999, 36(4), 390-398.
[7] Ferris M. C. and Munsion T. S., Interior-point methods for massive support vector machines, SIAM Journal of Optimization, 2002, 13(3),783-804.
[8] Horst R. and Thoai N. V., A new algorithm for solving the general quadratic programming problem, Computational Optimization and Applications, 1996, 5, 39-48.
[9] Khurana A. and Arora S.R., Cost-time constrained indefinite quadratic transportation problem, International Journal of Optimization: Theory, Methods and Applications, 2010, 2(3), 239-251.
[10] Martin A. D., Mathematical programming of portfolio selections. Management Science, 1955, 1(2), 152-166.
[11] Wolfe P., The simplex method for quadratic programming, Econometrica, 1959.

Appendix A (MATLAB code)

```
m=input('Enter the value of m: ');
n=input('Enter the value of n: ');
C=input('Enter the C matrix: ');
D=input('Enter the D matrix: ');
A=input('Enter the A matrix: ');
B=input('Enter the B matrix: ');

t=cputime
k=0
for i = 1:m
if (B(i) == 0)
k = k+1;
l(k)= i;
end
end
t1 =cputime-t
for j = 1:k
t2=cputime
r=0;
p=0;
q=0;
s=l(j)
for i = 1:n
if(A(s,i)==0)
r=r+1;
for f = 1:n
if(f==i)
T(f,r)=1;
else
T(f,r)=0;
end
end
elseif(A(s,i)>0)
p=p+1;
AP(p)=i;
else
q=q+1;
AN(q)=i;
end
end
g=r+(p*q)
for x = 1:p
for y = 1:q
r=r+1;
for z = 1:n
if (z==AP(x))
T(z,r)= -A(s,AN(y));
elseif(z==AN(y))
T(z,r)=A(s,AP(x));
else
T(z,r)=0;
```

```
   end
   end
   end
   end
   disp(['T' num2str(j) ' = ']);
   disp(T(1:n,1:g));
   C=C*T(1:n,1:g);
   D=T(1:n,1:g)'*D*T(1:n,1:g);
   A=A*T(1:n,1:g);
   if(j==1)
   H=T(1:n,1:g);
   else
   H=H*T(1:n,1:g);
   end
   n=r;
   t2=cputime-t2
   t1=t1+t2;
   disp(['processing time taken till step' num2str(j) ' = ']);
   disp(t1);
   disp(['A' num2str(j) ' = ']);
   disp(A);
   disp(['C' num2str(j) ' = ']);
   disp(C);
   disp(['D' num2str(j) ' = ']);
   disp(D);
   end
   disp('Matrix for converting back to original variables, H = ');
   disp(H)
   disp('Total processing time taken(doesnot include time taken to
display variables) = ');
   disp(t1);
```

Note: Here matrix H is the product of all transformation matrices and D equals matrix Q